51CTO WOT

World Of Tech 2024

WOT全球技术创新大会

2024

智启新纪
慧创万物

51CTO WOT

OceanBase 如何在同一套系统中做到交易与分析能力一体化

周跃跃
OceanBase 架构师

1、大数据场景下如何选择合适的数仓方案

2、分布式技术能为企业解决哪些问题

3、OceanBase OLAP 关键技术原理解决

4、更多 OLAP 功能

以快递行业为例

在快递流转中，物流公司需要实时掌握从用户下单到用户签收整个流程的快递运转情况，需要将网点产生的实时物流信息写入数据库中并动态分析业务状况，发现每一个环节可能出现的问题以及快速解决，提升运营效率，提高用户体验。在特殊节假日流量高峰期，日均产生的数据量亿级别，同时伴随着分析压力。

核心是：存、算（实时）

在该类场景下， Hive+ Spark 可能成为最佳方案，不过问题在于：

- 数据延迟高
  - 数据导入延迟
    - 所有数据通过 kaKfa 定时导入至 Hive 数据仓库，无法做到实时更新，数据时延超过10分钟以上
  - 延迟查询
    - 从 kafka 实时写入 Hive，就要把批量写入时间设置成很短，会产生很多小文件，也没法做到实时查询

- 查询耗时长
  - 使用 Spark 读取 Hive 进行数据分析统计时，进行一次上亿数据的统计需要3分钟以上

- 成本高
  - 使用 Spark 进行数据定期导入与分析统计，消耗大数据集群CPU、内存资源较高，同一时间任务太多时需要排队执行；同时组件多。

数据库能不能解决？

对比项	Hive	传统数据库
数据插入	支持批量导入	支持单条和批量导入
数据更新	不支持	支持
索引	有限索引功能，不像RDBMS有键的概念，可在某些列上建索引，加速一些查询操作。创建的索引数据会被保存在另外的表中	支持
分区列	支持，Hive表示分区形式进行组织的，根据“分区列”的值对表进行粗略划分，加快数据的查询速度	支持，提供分区功能来改善大型表以及具有各种访问模式的表的可伸缩性、可管理性、以及提高数据库效率
执行延迟	高，构建在HDFS和MR之上，比传统数据库延迟要高	低，传统SQL语句的延迟一般少于1秒，而HSQL语句延迟可达分钟级
扩展性	好，基于Hadoop集群，有很好的横向扩展性	有限，RDBMS非分布式，横向扩展（分布式添加节点）难实现，纵向（扩展内存，CPU等）也很有限

传统数据库方案：
1、高并发写入与更新
2、索引能力
3、扩展性有限
4、分析能力有限

1、大数据场景下如何选择合适的数仓方案

2、分布式技术能为企业解决哪些问题

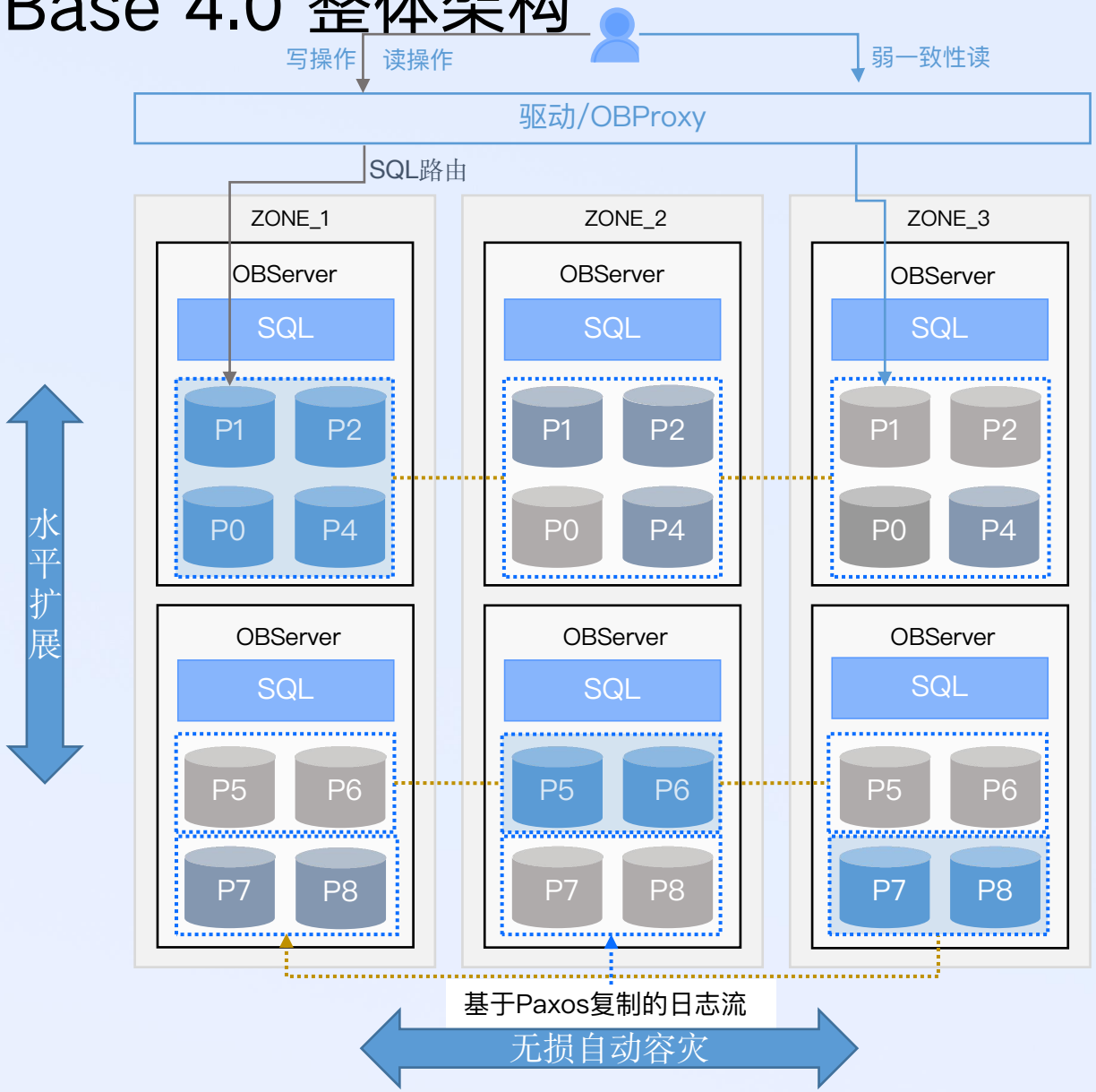
3、OceanBase OLAP 关键技术原理解决

4、更多 OLAP 功能

OceanBase 功能特性

- 完全具备关系型数据库的SQL、存储、事务能力
- TPC-C 7.07亿tpmC 世界第一
  - TP 能力已经在多家客户、用户场景实践验证
- 水平和垂直扩展，自动负载均衡，弹性扩缩容
  - 扩展能力如何？
- 同一套引擎同时支持 OLTP 和 OLAP 混合负载
  - OLAP 能力如何？

OceanBase 4.0 整体架构



对等节点

- 无共享集群
- OBServer包含SQL、存储、事务

高可扩展性

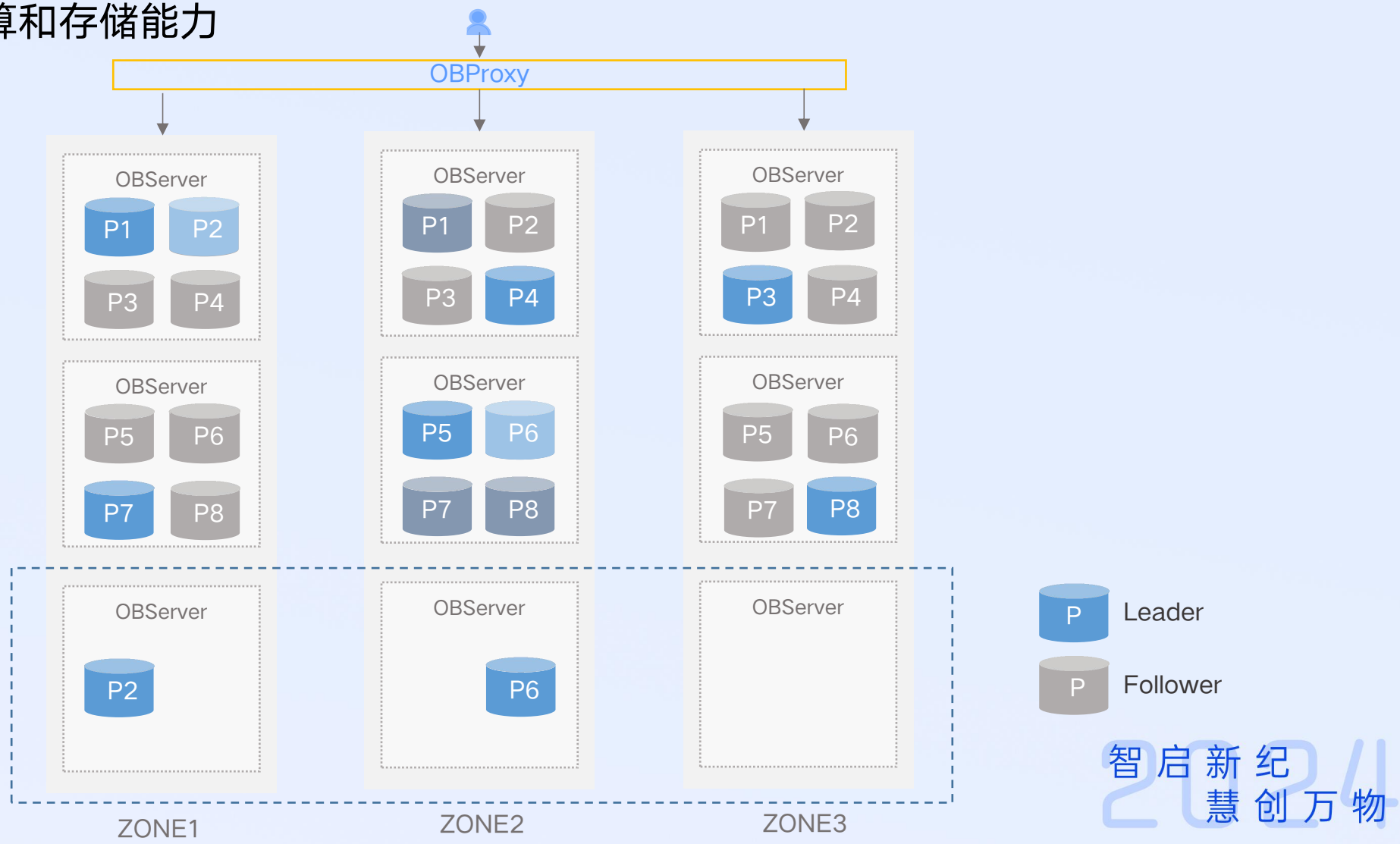
- 按分区做数据分片扩展
- 多Zone多活扩展

单机分布式一体化

- 日志流：数据库的所有变更
- 多个分区可共用一个日志流
- 单机内无分布式事务
- 低时延分布式处理技术

垂直扩展

每个节点都具备计算和存储能力



水平扩展

- 集群级别追加zone
- 集群级别追加zone
- 数据自动进行复制
  - ✓ 数据同步速度 > 500MB/s
- 自动选出Leader
  - ✓ 根据zone的优先级
  - ✓ 无需停服务

P Leader

P Follower

蚂蚁现状

以下数据来自于**实际**生产系统

6100
万次/秒
数据库峰值处理能力

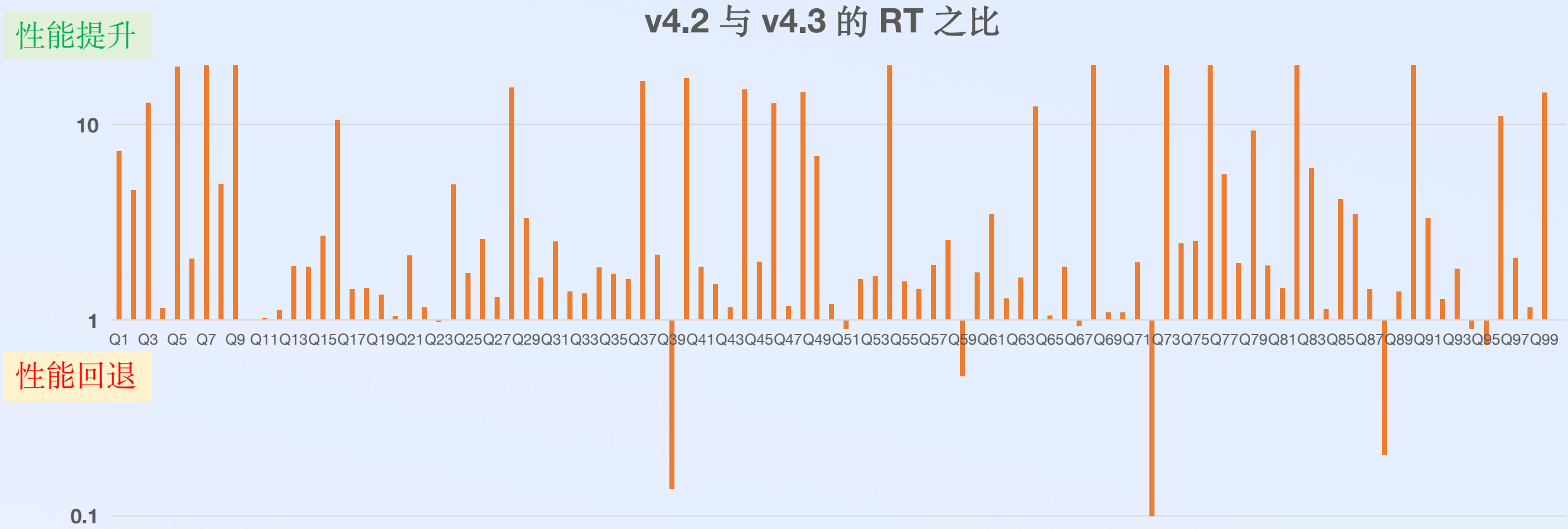
>1000
台
单集群节点数

>6
PB
单库存储容量

>3200
亿行
单表行数

RPO=0,
RTO<8
秒
少数副本故障时

OLAP 能力：不同版本之间性能提升(TPC-DS 1T)最大 53%



V3.2.4	V4.1.0	V4.2.0	V4.3.0 (列存)
4195.18s	2293.60s	**1519.41s**	**717.62**

性能逐步提升：45%->35%->53%

OLAP 能力：对比 ClickHouse 以及某云数据库

CK 耗时（秒）(集群规格 32C 256G)	OB 4.3 (列存) 耗时（秒）(租户规格 24C 120G)	OB 4.3 (列存) 耗时（秒）(租户规格 30C 180G)	OB 4.2.1(行存) 耗时（秒）(租户规格 58C 356G)
5.05	3.049	2.593	2.547
13.97	7.667	6.113	6.398
35.96	13.555	10.955	11.098
4.42	6.341	5.032	10.453
20.58	22.637	16.699	39.162
37.94	36.977	28.818	62.367

对比 CK:
在资源规格接近，大部分情况下 OB 4.3 耗时为 CK 1/ 2 甚至于 1/3

某云数据库 耗时（秒）(集群规格 24C 192G)	OB 4.3 (列存) 耗时（秒）(租户规格 24C 120G)	OB 4.3 (列存) 耗时（秒）(租户规格 30C 180G)	OB 4.2.1(行存) 耗时（秒）(租户规格 58C 356G)
1058	289	144	365
291	227	147	107
22	6	4.5	19
124	6	5.5	61
1234	291	231	739
1024	293	224	715
1190	320	190	208
54	11	11	30

对比 某云数据库:
在资源规格接近时，4.3 OB 整体耗时低于某云，甚至为某云 1/5 且随着资源增加，耗时不断减少

1、大数据场景下如何选择合适的数仓方案

2、分布式技术能为企业解决哪些问题

3、OceanBase OLAP 关键技术原理解决

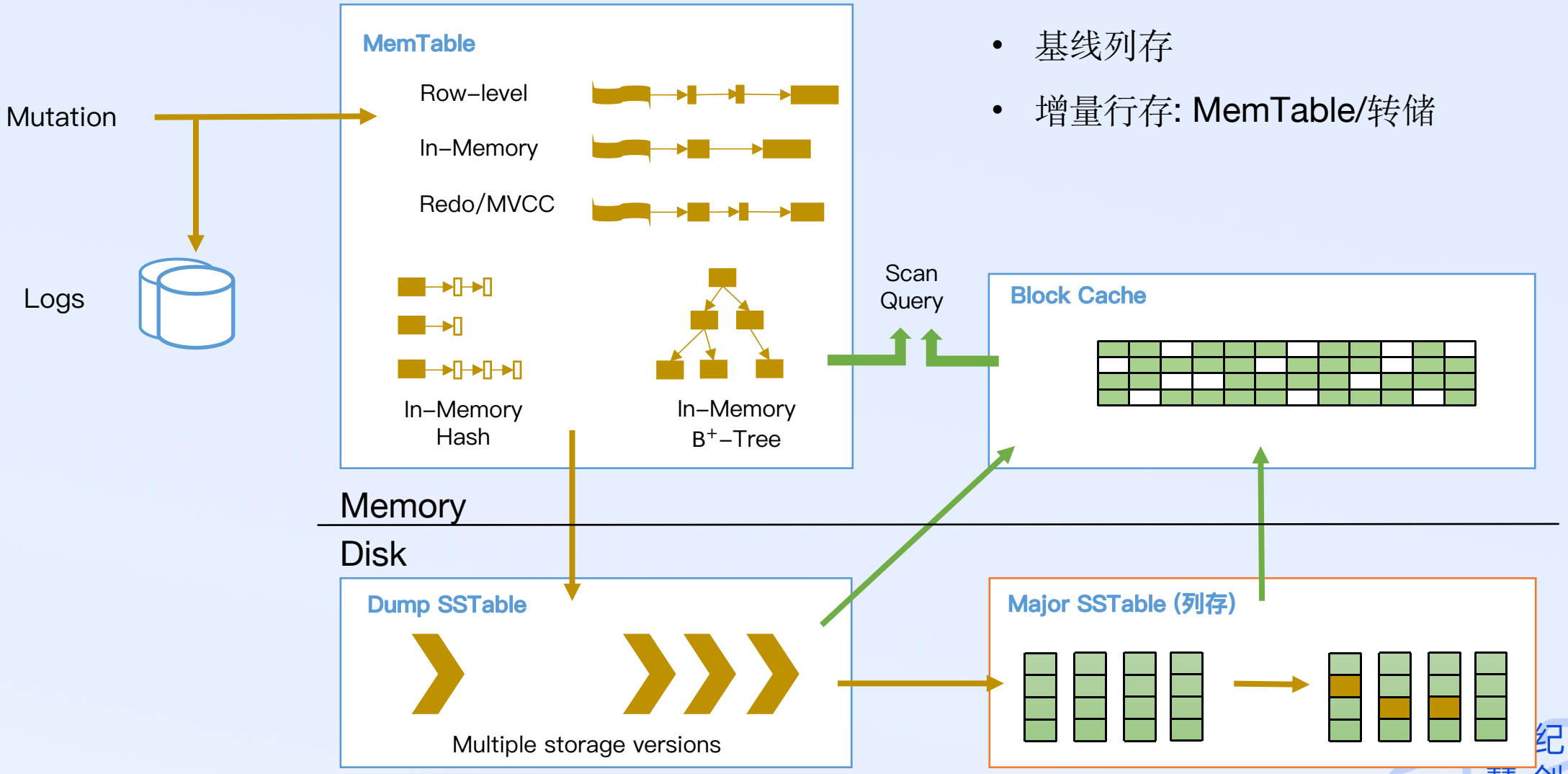
4、更多 OLAP 功能

算的更快

基于分布式架构，企业级查询优化器 & 向量执行引擎 2.0

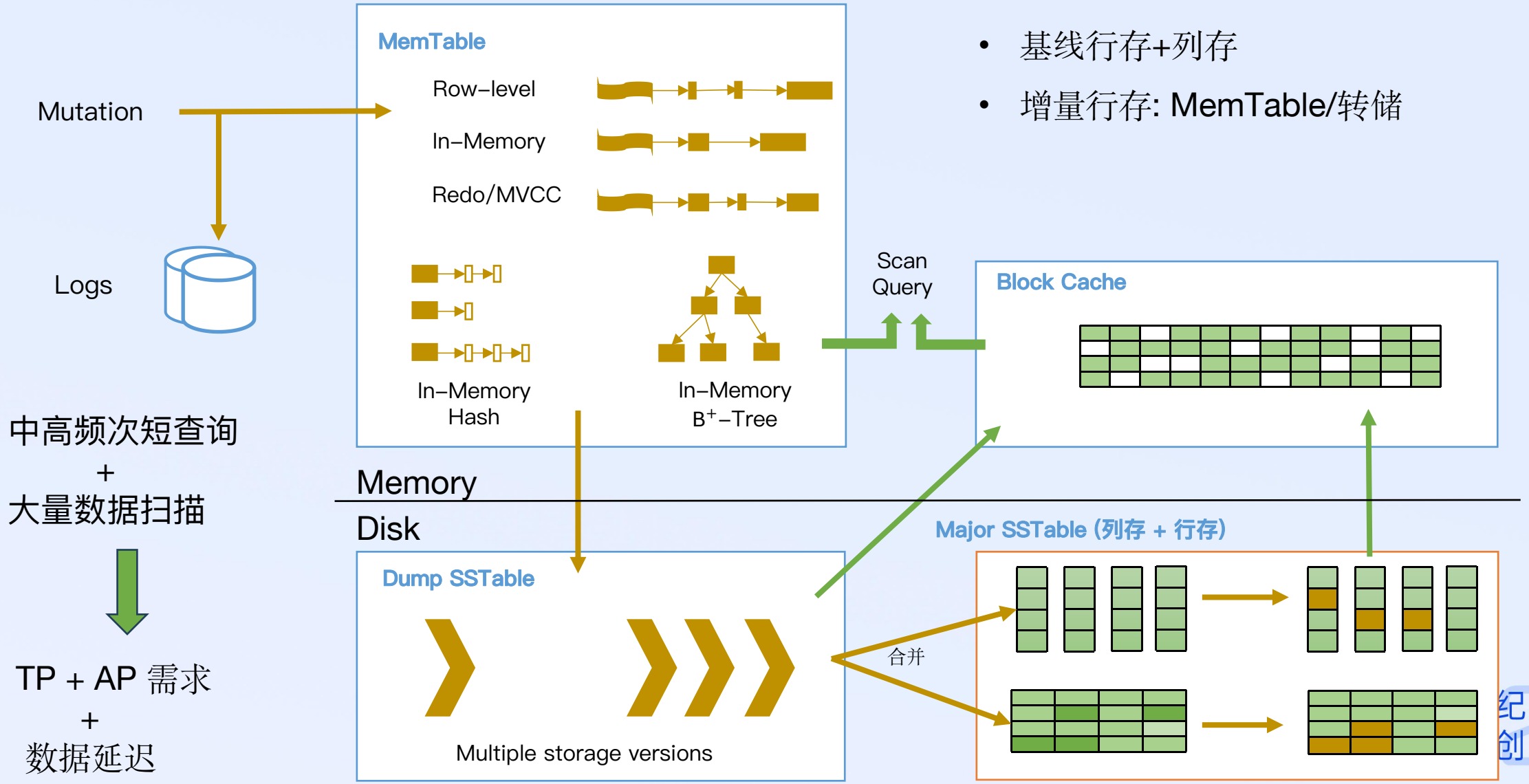
向量化引擎 2.0 更进一步挖掘了SIMD 计算、特化实现及按批访问和处理数据等带来的性能提升，更快地处理AP场景的SQL请求

列存表的存储引擎架构



- 基线列存
- 增量行存: MemTable/转储

行列混合表的存储引擎架构



- 基线行存+列存
- 增量行存: MemTable/转储

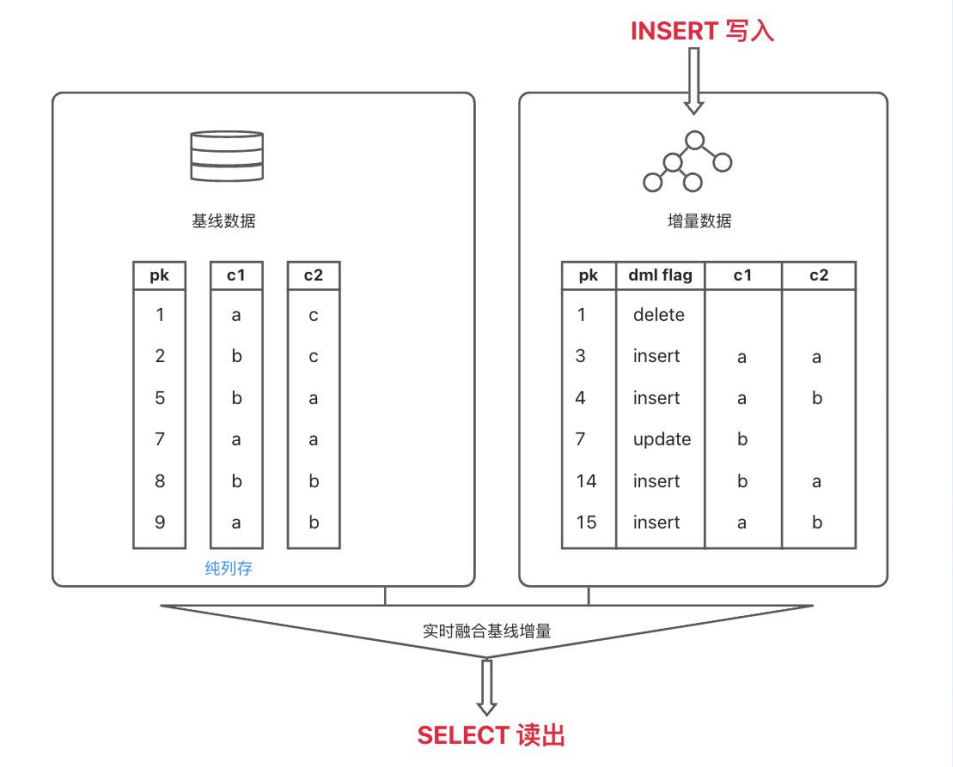
中高频次短查询
+
大量数据扫描

TP + AP 需求
+
数据延迟

实时性 & 自由转换

Session A 更新:**insert into t1 values (...)**

Session B 分析:**select count(*) from t1;**



相比于纯列存产品，实时性更高!

```
create table t1(
   pk int,
   c1 varchar(1),
   c2 varchar(1)
) with column group (all columns, each column);
```

pk	c1	c2
1	a	c
2	b	c
5	b	a
7	a	a
8	b	b
9	a	b

行存冗余

pk	c1	c2
1	a	c
2	b	c
5	b	a
7	a	a
8	b	b
9	a	b

纯列存

语法说明

- all columns：一个 column group 里包含表里所有的列，等价于 v4.2 里的行格式存储
- each column：表中的每一列分别使用列格式存储

更多使用姿势

列存索引：索引表的结构是列存格式

逻辑结构

pk	c1	c2
1	a	c
2	b	c
5	b	a
7	a	a
8	b	b
9	a	b

主表（行存）

c2
c
c
a
a
b
b

pk
1
2
5
7
8
9

索引表（列存）

应用场景：将行存表中的若干列转化成列存，用于分析。

行存索引

逻辑结构

pk	c1	c2
1	a	c
2	b	c
5	b	a
7	a	a
8	b	b
9	a	b

纯列存

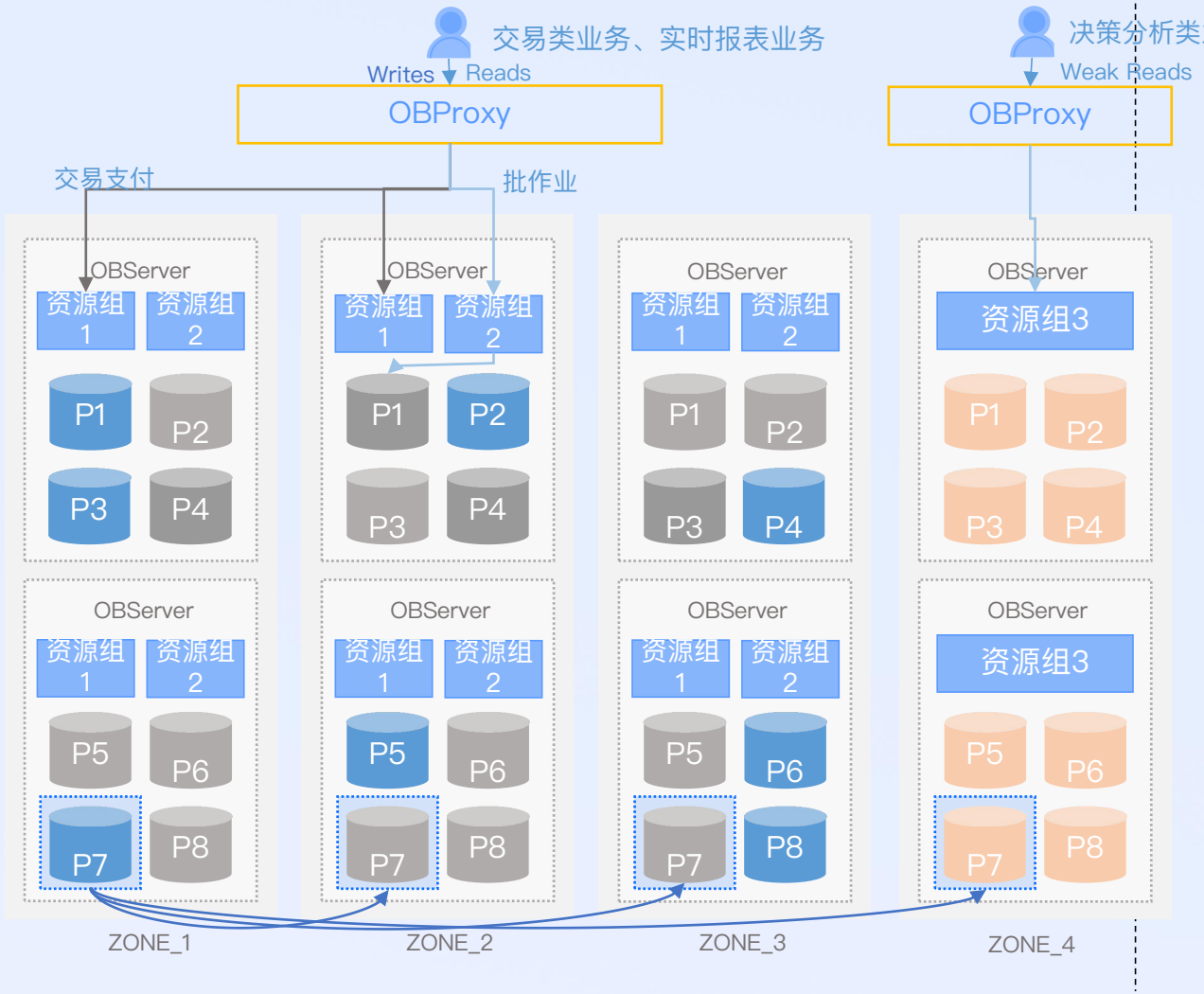
主表（列存）

pk	c1
1	a
2	b
5	b
7	a
8	b
9	a

索引表（行存）

将列存表中的若干列转化成行存，用于高性能点查。

混合负载的读写分离、资源隔离

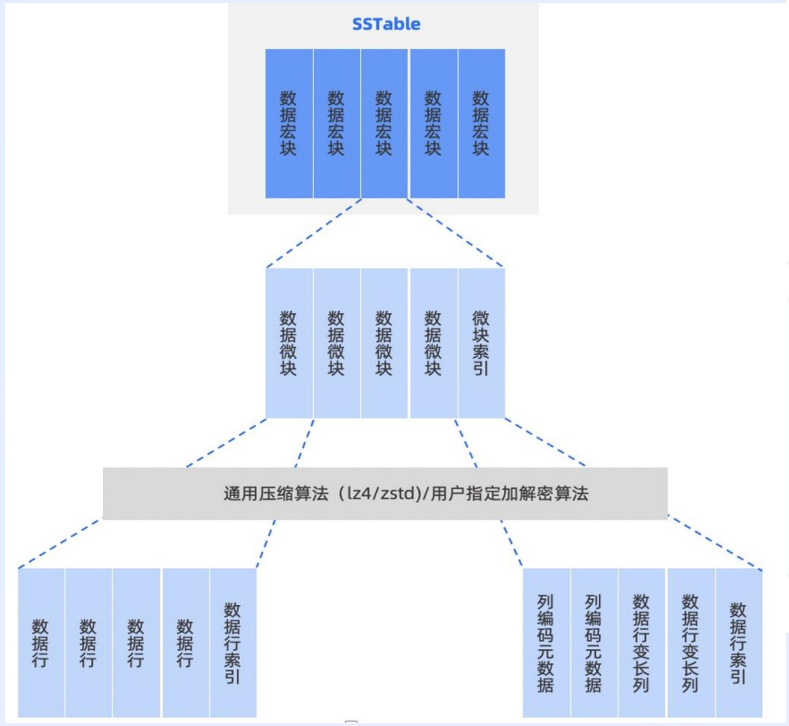


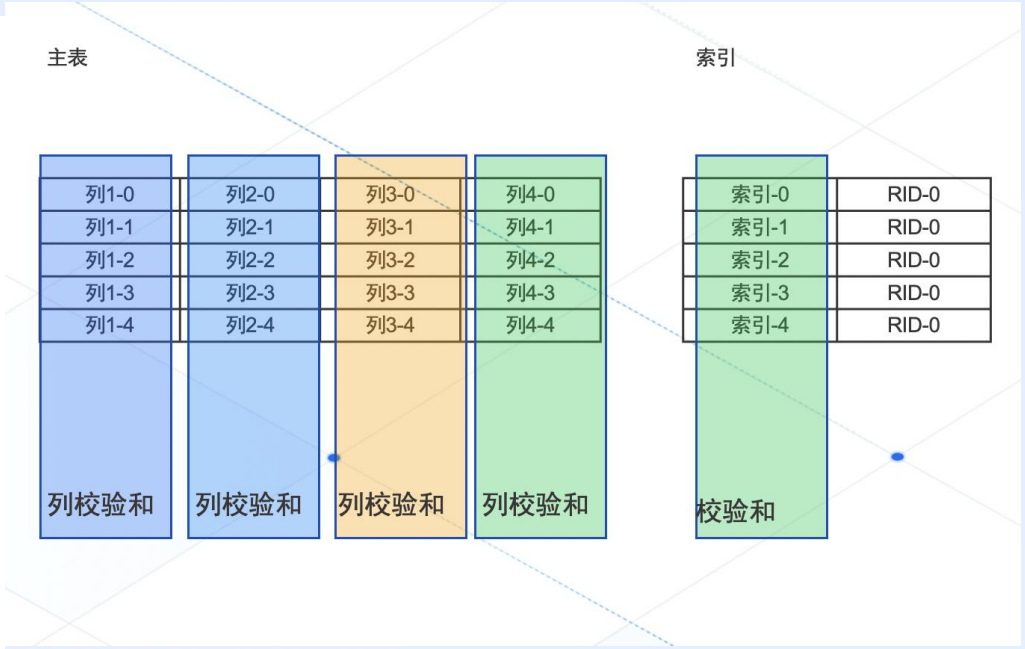
灵活的隔离机制

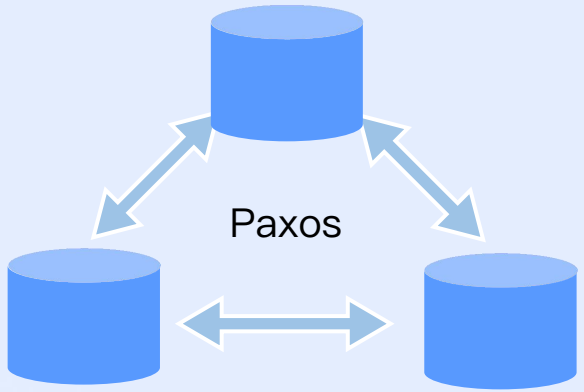
- 物理隔离
  - 弱一致性读读写分离
  - 多Zone读写分离
- 混合负载
  - 基于cgroup的资源组
    - 用户名匹配
    - SQL语句级匹配
  - 大查询自动隔离
    - 独立大查询队列

极致高可用-基于 paxos 的多副本架构

- 基于 paxos 的多数派投票协议，默认三副本，包括一个 leader 两个 follower

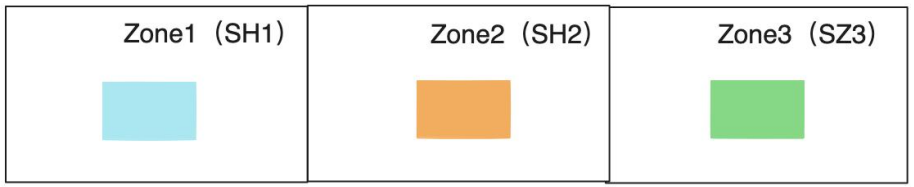




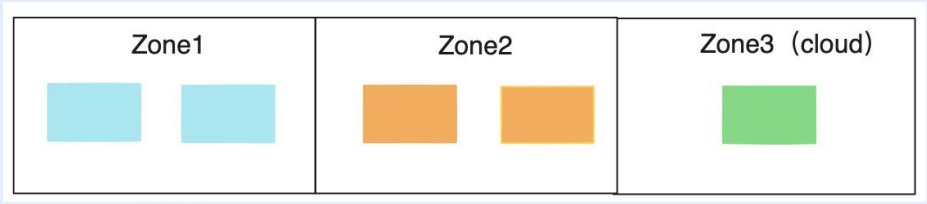


- SSTable 累计行校验和
- SSTable 列校验和
- 合并时:
  - 索引列 列校验和和主表列的列校验和 进行比较
  - 副本之间的行校验和 和 列校验和 进行比较

极致高可用-基于架构的高可用解决方案

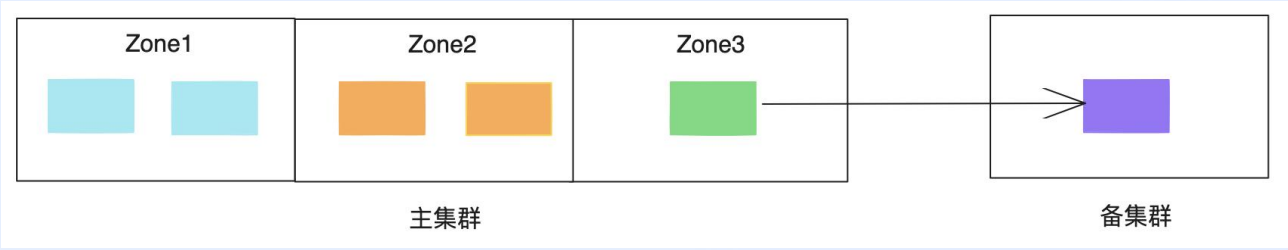


两地三中心

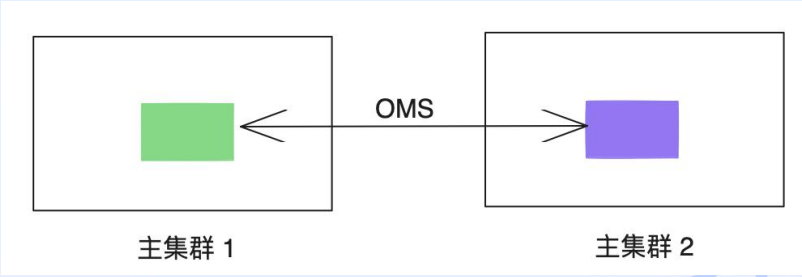


混合云

三地五中心



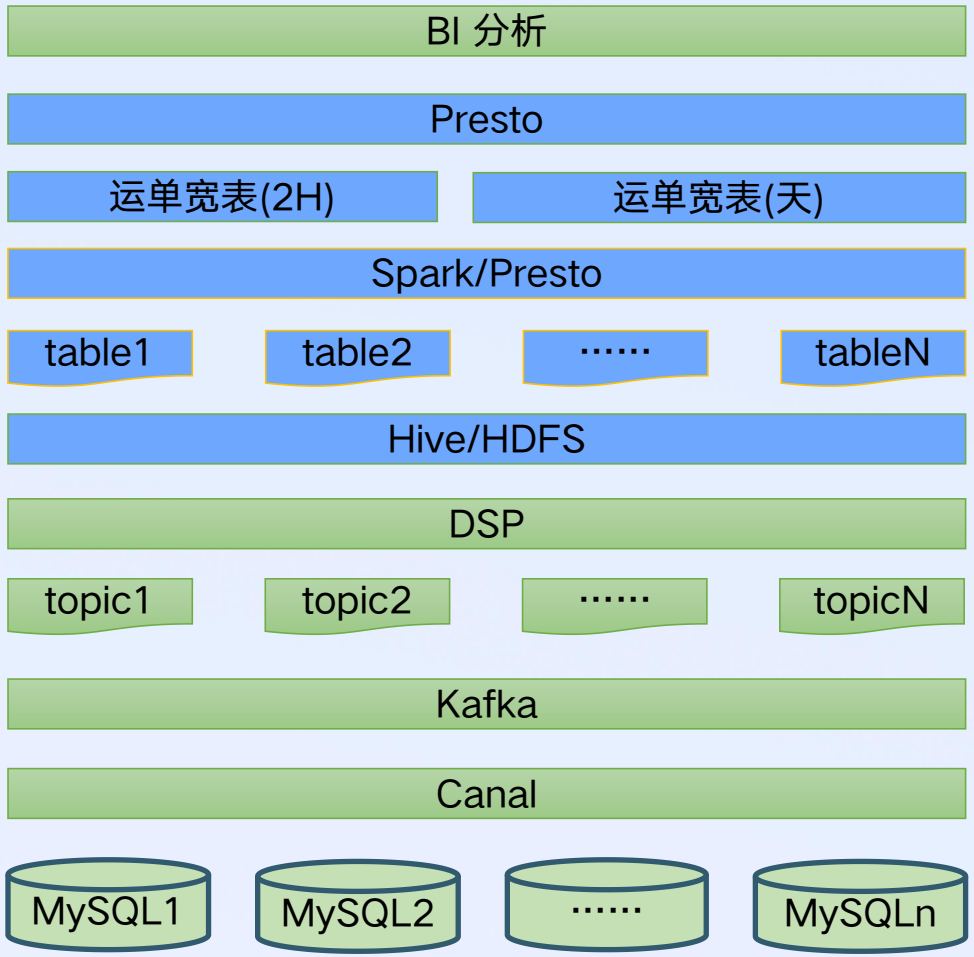
主备库

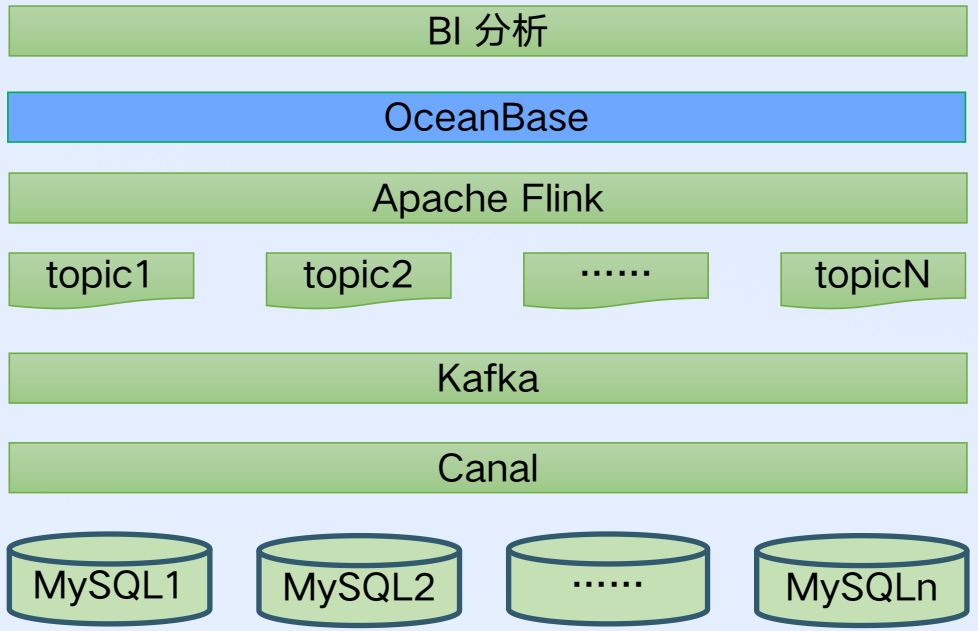


双主

实时分析场景下如何使用 OceanBase？

运单中心





优化前：

数据时效差：2 小时
分析性能差：1-10s

优化后：

- 数据时效：<2s
- 分析性能：<4s
- 成本降低：50%

性能不弱于 AP 产品
- AP能力不断提升
- 数据链路短

极致的扩展性
- 解决存算瓶颈问题

OceanBase

架构更简洁
- 一套替换多个组件
- 管理方便

稳定性有保证
- 体验关系型数据库不宕机能力

总结

1、大数据场景下如何选择合适的数仓方案

2、分布式技术能为企业解决哪些问题

3、OceanBase OLAP 关键技术原理解决

4、更多 OLAP 功能

物化视图

	MV刷新策略	MV刷新方法	单表聚合	多表关联	查询改写	基表
非实时物化视图	异步	定时全量刷新 手动全量刷新	Yes	Yes	Yes	普通表 已有物化视图 普通视图 外表
	异步	定时增量刷新 手动增量刷新	Yes	Yes(五表内连接)	Yes	普通表
实时物化视图	异步	定时全量刷新 手动全量刷新	Yes	Yes(五表内连接)	Yes	普通表
	异步	定时增量刷新 手动增量刷新	Yes	Yes(五表内连接)	Yes	普通表
	同步 (on commit)	实时生效	Yes	Yes	Yes	普通表

4.3.0 4.3.1 4.3.2 暂无规划

旁路导入

OceanBase 提供旁路导入方案，解决过去导入性能不足、稳定性差的难题。 数据导入性能提升3~10倍，导入稳定性极大提升。

典型场景介绍

- 批量数据导入（如 PoC 等）
- 内存不足时可能被写入限流，导致导入时间变长
- Memtable 转储不够快时，可能报 Out Of Memory，导入失败

方案价值

- 绕开 Memtable 直接写存储，减少不必要开销，提升写入性能
- 绕开 Memtable，租户内存大小与数据导入量解耦
- 数据写入效率大幅提升

4.3.0 旁路导入最佳实践:

1. 空表，一次性导入全部数据，性能最佳

```
LOAD DATA /*+ direct(need_sort, max_error_allowed) */
INFILE 'file_name' IN TO TABLE table_name;

insert /*+ append enable_parallel_dml parallel(16) */ into
to_table select * from from_table
```

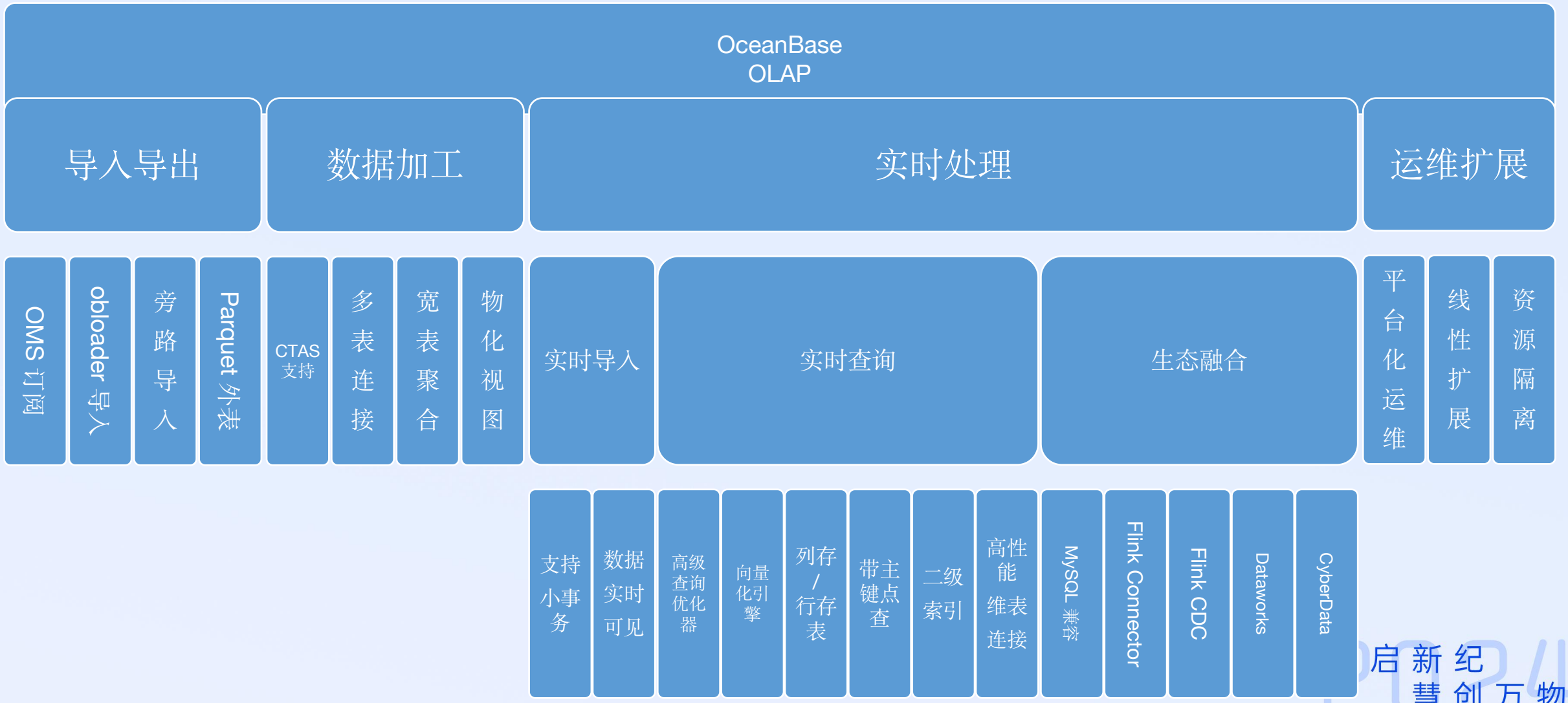
lineitem(100G)在单机c6a.4xlarge导入性能对比

数据库	表的schema	load time
OB （旁路导入）	堆表	737s
OB （非旁路）	堆表	5160s
OB （旁路导入）	索引组织表	1402s
OB （非旁路）	索引组织表	4920s

内核 Roadmap — 4.3.x

	v4.3.0（3月底）	v4.3.1（5 月中）	v4.3.2（7 月中）	v4.3.3（9 月底，GA）
云上架时间	2024.03（POC+白名单）	2024.05（POC+白名单）	2024.07(POC+白名单）	2024.09 GA
功能增强	全量旁路导入支持列存 物化视图 Basic 租户克隆	分区交换 增量旁路导入 Basic 物化视图 Enhanced	增量旁路导入 Enhanced 外表支持走JDBC驱动访问外部库（如 MC、Hive 等） 物化视图 gby 改写	列存副本、Vector size 分区、行级 TTL Tunnel 访问 ODPS/MC 外表支持走 HDFS 驱动访问文件 增量旁路导入 Enhanced (index, 主键表+lob)
兼容性增强		全文索引 JSON 多值索引	Bitmap Basic、Parquet 作外表、JSON多值索引 Enhanced Insert Overwrite Basic	Insert overwrite Enhanced (partition)、ORC 作外表、Array 全文索引 Enhanced
性能提升	列存 全新向量化引擎 PDML 事务优化 DDL 临时结果空间优化			性能基本追平头部AP系统
产品形态	OLAP 引擎			向量计算

OceanBase 实时分析能力支撑



51CTO WOT

谢 谢 观 看

THANKS

智启新纪
慧创万物
2024